techspeak

BY JAY M. JACOBSMEYER, P.E.

Demystifying error-control coding

It's the key to designing error-free communications systems. But what is it?

Part 1 of 4

Many of us have heard the term error-control coding in connection with modern land mobile radio systems, but few of us understand what it is, what it does and why we should be concerned about it. Old timers are forewarned: All LMRs eventually will be digital, and all digital radios use error-control coding. Terms like Golay code, Hamming code, CRC and interleaving will become popular topics at cocktail parties. Don't get left out of the conversation; learn about error-control codes.

This article begins a series that will examine what error-control codes do, how they work, and when they can be used successfully.

Error-control coding is a discipline under the branch of applied mathematics called Information Theory, discovered by Claude Shannon in 1948 [1]. Prior to this discovery, conventional wisdom said that channel noise prevented error-free communications. Shannon proved otherwise when he showed that channel noise limits the transmission rate, not the error probability.

Shannon showed that every communications channel has a capacity, C (measured in bits per second), and as long as the transmission rate, R (also in bits per second), is less than C, it is possible to design a virtually error-free communications system using error-control codes. Shannon's contribution was to prove the existence of such codes. He did not tell us how to find them.

After the publication of Shannon's famous paper, researchers scrambled to find codes that would produce the very small probability of error that he predicted. Progress was disappointing in the 1950s when only a few weak codes were found.

In the 1960s, the field split between the algebraists, who concentrated on a class of codes called block codes, and the probabilists, who were concerned with understanding encoding and decoding as a random process. The probabilists eventually discovered a second class of codes, called convolutional codes, and designed powerful decoders for them.

In the 1970s, the two research paths merged, and several efficient decoding algorithms were developed. With the advent of inexpensive microelectronics, decoders finally became practical and in 1981, the entertainment industry adopted a very powerful error control scheme for the new CD player [2]. Today, error-control coding in its many forms is used in almost every new communications system, including the Association of Public-Safety Communications Officials' Project 25 standard.

Digital communications systems often are conceptualized as shown in Figure 1. The following paragraphs describe the elements of Figure 1 and define other terms common to error-control coding.

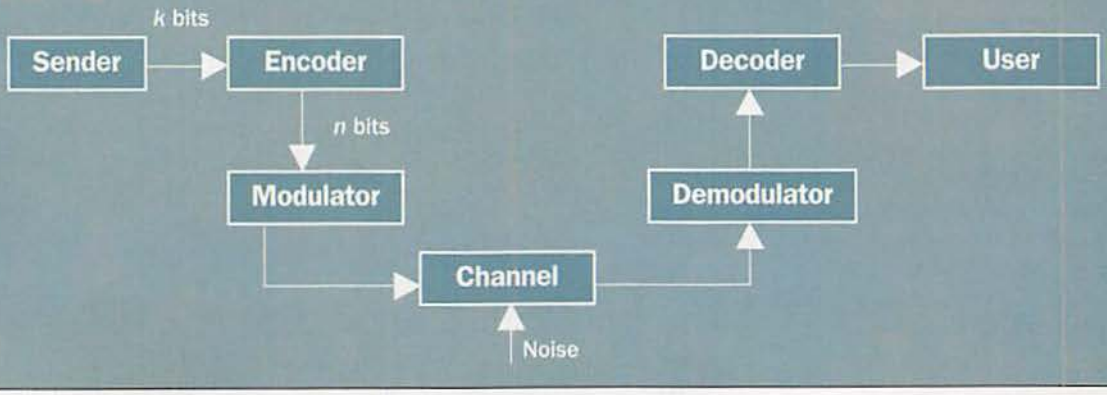
■ **Encoder and decoder:** The encoder adds redundant bits to the sender's bit stream to create a code word. The decoder uses the redundant bits to detect and/or correct as many bit errors as the particular error-control code will allow. For our purposes, encoding and decoding refers to channel coding as opposed to source coding. (Source coding prepares the original information for transmission; the vocoder, or voice-encoder, is one example of source coding.)

■ **Modulator and demodulator:** The modulator transforms the output of the encoder, which is digital, into a format suitable for the channel, which is usually analog (e.g., a radio channel). The demodulator attempts to recover the correct channel symbol in the presence of noise.

When the wrong symbol is selected, the decoder tries to correct any errors that result. Some demodulators make soft decisions, meaning that the de-

Continued on page 34

Figure 1



Continued from page 30

modulator does not attempt to match the received signal to one of the allowed symbols. Instead, it matches the noisy sample to a larger set of discrete symbols and sends it to the decoder where the heavy lifting is done.

All LMRs eventually will be digital, and all digital radios use error-control coding.

■ **Communications channel:** This part of the communication system introduces errors. The channel can be radio, twisted wire pair, coaxial cable, fiber-optic cable, magnetic tape, optical discs or any other noisy medium.

■ **Error-control code:** The set of code words used with an encoder and decoder to detect errors, correct errors, or both detect and correct errors.

■ **Bit-error rate (BER):** The probability of bit error often is the figure of merit for an error-control code. We want to keep this number small, typically less than 10^{-4} for data and less than 10^{-3} for digital voice. BER is a useful indicator of system performance on an independent error channel, but it has little meaning on bursty, or dependent, error channels.

■ **Message-error rate:** This probability of message error is sometimes called frame error rate. This may be a more appropriate figure of merit because the smart user wants all of his or her messages to be error-free and couldn't care less about the BER.

■ **Undetected message error rate (UMER):** This is the probability that the error detection decoder fails and an errored message (code word) slips through undetected. This event happens when the error pattern introduced by the channel is such that the transmitted code word is converted into another valid code word. The decoder can't tell the difference and must conclude that the message is error-free. Practical error detection codes ensure that the UMER is very small, often less than 10^{-16}.

■ **Random errors:** These errors occur independently. This type of error occurs on channels that are impaired solely by thermal (Gaussian) noise. Independent-error channels also are called memory-less channels because knowledge of previous channel symbols adds nothing to our knowledge of the current channel symbol.

■ **Burst errors:** These errors are not independent. For example, channels with deep fades experience errors that occur in bursts. Because the fades result in consecutive bits that are more likely to be in error, the errors are usually considered dependent rather than independent. In contrast to independent-error channels, burst-error channels have memory.

Equation 1

$$E_b = \frac{P_t}{R}$$

■ **Energy per bit:** This refers to the amount of energy contained in one information bit. This is not a parameter that can be measured by a meter, but it can be derived from other known parameters. Energy per bit (E_b) is important because almost all channel impairments can be overcome by increasing it. Energy per bit (in joules) is related to transmitter power P_t (in watts and bit rate R (in bits per second), as shown in Equation 1.

If transmit power is fixed, the energy per bit can be increased by lowering the bit rate. This is why lower bit rates are considered more robust. The required energy per bit to maintain reliable communications can be decreased through error-control coding, as we shall see in the next article in this series.

■ **Coding gain:** This refers to the difference in decibels (dB) in the signal-to-noise ratio required to maintain reliable communications after coding is employed. Signal-to-noise ratio is usually represented as E_b/N_0, where N_0 is the noise power spectral density measured in watts/Hertz (joules). For example, let's say a communications system requires an E_b/N_0 of 12 dB to maintain a BER of 10^{-5}, but after coding it requires only 9 dB to maintain the same BER. In that case, the coding gain is 12 dB - 9 dB = 3 dB. (Recall that dB = 10 log10 X, where X is a ratio of powers or energies.)

■ **Code rate:** Consider an encoder that takes k information bits and adds r redundant bits (also called parity bits) for a total of $n = k + r$ bits per code word. The code rate is the fraction k/n, and the code is called an (n, k) error-control code. The added parity bits are a burden (i.e., overhead) to the communications system, so the system designer often chooses a code for its ability to achieve high coding gain with few parity bits.

Next month: What coding can and cannot do. ■

Jay Jacobsmeyer is president of Pericle Communications Co., a consulting engineering firm located in Colorado Springs, Colo. He holds bachelor's and master's degrees in Electrical Engineering from Virginia Tech and Cornell University, respectively, and has more than 20 years experience as a radio frequency engineer.

References:

[1] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, pp. 379-423, 1948.

[2] F. Guterl, "Compact Disc," IEEE Spectrum, vol. 25, No. 11, pp. 102-108, 1988.

"Moducom's system lets you put the two screens side by side to create one large screen and your mouse moves back and forth between the screens."
—MIKE SIKORSKI, ORLAND CENTRAL DISPATCH CENTER

ing an emergency call because the dispatcher knows where everything is on the screen. Another plus is the system's cloning feature, which allows administrators to rename and save screens rather than continually creating them from scratch.

These benefits are dwarfed by the system's dual-screen capability, which is a major plus given the amount of traffic and number of agencies handled by the Orland Central Dispatch Center, Sikorski said. The center currently provides dispatch services for the Orland Park Fire Department and six other surrounding departments and handles extra alarms for a total of 35 departments.

In addition to 911 call data, screens also display computer-aided dispatch and mapping information and let dispatchers control various building lock boxes—which can be opened by a radio tone, giving firefighters entry into the building without them having to carry a key for every business or municipal building in town—fire station doors and radio towers, letting them place the towers in and out of service as needed. The screens also display information that tells the dispatchers which radio frequencies are in use and how they're being used at any given moment.

"There's a lot to look at," Sikorski said. "It would be impossible to work it all onto one screen. Moducom's system lets you put the two screens side by-side to create one large screen, and your mouse moves back and forth between the screens."

Machuta agreed. "The typical reason that anyone would go to dual-screen is the amount of information," he said. "If they have a lot of information—and a lot of departments that they dispatch to—or they want to have other technology on the screen at the same time, the density of the amount of information on one screen makes it too busy for someone to be able to sort out and manage it, especially in a concentrated environment. You want your icons and other indicators large enough so that you can press those without disturbing two or three other processes."

Moducom's dual-screen console

There are several ways that the dual screens can be configured, but Machuta advised departments the size and scope of the Orland Central Dispatch Center to place information from their primary radio systems on one screen.

"These would be the systems they talk to all the time," he said. "You could have 10 to 15 radio systems on the primary screen, and it wouldn't be too busy to see what's going on."

Machuta suggested using the secondary screen to display surrounding departments and the secondary dispatch centers with which they communicate on a regular basis, as well as the local telephone interface.

According to J&L Electric's Patrizi, seven Moducom dispatch consoles currently are deployed in the Chicago metropolitan area, including dual-screen systems at Orland Central Dispatch, Countryside Police Department and Hickory Hills Police Department. The largest system in the area is scheduled to be online in February at Southwest Central Dispatch in Palos Heights, Ill., which serves agencies in three counties. The half-million dollar system provides six positions and is expandable to 10, Patrizi said.

Single-screen systems are scheduled for deployment in the villages of Northfield and Riverside in early spring, joining similar systems at Orland Park Police and in the villages of Stone Park and Alsip.

They might want to wait just a little longer, as Moducom is planning a system upgrade that will add a radio-logging capability "very shortly," said Don Poysa, regional sales manager.

"Currently, the system has a server that permanently stores and logs all of the telephone calls from the 911 and admin lines and provides full reporting capability," he said. "We don't have that on the radio side, and that's what we're going to be adding in a year or so.

"It's probably not a big selling point because most of the agencies already have [radio] logging recorders, but we've had some people tell us it would be nice to have that [in a single system] so they wouldn't have to go out and buy one because they cost from $20,000 to $25,000." ■

BY JAY M. JACOBSMEYER, P.E.

What error-control coding can do

However, the coding's limitations also must be realized

Part 2 of 4

In the November 2005 *MRT*, we defined terms and introduced basic error-control coding concepts. This month we will cover some of the applications and limitations of codes and the theory behind their operation.

The traditional role for error-control coding was to make a troublesome channel acceptable by lowering the frequency of error events. The error events could be bit errors, message errors or undetected errors. While reducing the occurrence of undetected errors was one of the first uses of error-control coding, today's error-detection codes are so effective that the occurrence of undetected errors is, for all practical purposes, eliminated. Today, the role of error-control coding as expanded to include many new applications, including:

■ **Reduce the cost of communications systems:** Transmitter power is expensive, especially on satellite transponders. Coding can reduce the satellite's power needs because messages received at close to the thermal noise level can still be recovered correctly.

■ **Eliminate interference:** As the electromagnetic spectrum becomes more crowded with man-made signals, error-control coding will mitigate the effects of unintentional interference.

Despite these new uses of error-control coding, there are limits to what coding can do. On the Gaussian noise channel, for example, Shannon's capacity formula sets a lower limit on the signal-to-noise ratio that we must achieve to maintain reliable communications. For strictly power-limited (unlimited bandwidth) channels, Shannon's lower threshold can be expressed by E_b/N_0 equals 0.69, or -1.6 decibels (dB) [1]. In other words, we must maintain an E_b/N_0 of at least -1.6 dB to ensure reliable communications, no matter how powerful an error-control code we use.

Many channels, like the land mobile radio channel, are bandwidth limited. For bandwidth-limited channels with Gaussian noise, Shannon's capacity formula can be written as Equation 1 where r is the spectral bit rate in bits/s/Hz [2].

For example, consider a bandwidth-limited channel operating with uncoded quadrature phase shift keying (2 bits/symbol and a maximum spectral bit rate of r=2 bit/s/Hz) and a required bit error rate (BER) of 10^{-5}. Without coding, this communications system requires an E_b/N_0 of 9.6 dB [2]. Shannon's formula says that to maintain reliable communications at an arbitrarily low BER, we must maintain (for r=2 bits/s/Hz) an E_b/N_0 of at least 1.5, or 1.8 dB.

Therefore, if we need to lower the required E_b/N_0 by more than 7.8 dB, coding can't do it. We must resort to other measures, such as increasing transmitter power. In practice, the situation is worse because no practical code achieves Shannon's lower threshold. Until recently, a more realistic coding gain for this example was 3 dB rather than 7.8 dB. The invention of turbo codes in 1993 allows us to nearly achieve Shannon's coding limit, but at the price of delay [3]. Consequently, turbo codes are good choices for data channels, but they are impractical for two-way voice channels.

A full understanding of the structure and performance of error-control codes requires a foundation in modern algebra and probability theory, which is beyond the scope of this column. Instead, I'll appeal to your intuition and common sense. Let's begin by showing how the encoder and decoder work for binary block codes.

Equation 1

$$\frac{E_b}{N_0} \geq \frac{2^r - 1}{r}$$

The block encoder takes a block of k bits and replaces it with an n-bit codeword (n is bigger than k). For a binary code, there are 2^k possible codewords in the codebook. The channel introduces errors and the received word can be any one of 2^n n-bit words of which only 2^k are valid codewords. The job of the decoder is to find the codeword that is closest to the received n-bit word. How a practical decoder does this is more than we can cover here, but our examples will use a brute force look-up table method. The decoding spheres represented in Figure 1 on page 35 will be used to illustrate the decoding process.

In Figure 1, each valid codeword is represented by a point surrounded by a sphere of radius t, where t is the number of errors that the code can correct [2]. Note that codewords A and B of Figure 1 are separated by a distance d_{min}, called the minimum distance of the code. The

minimum distance of a code is defined as the smallest number of places that any two codewords in the codebook differ. Usually, codes with large minimum distance are preferred because they can detect and correct more errors.

Let's first consider a decoder that can only detect errors, not correct them.

■ **Error detection only:** The minimum distance of a code is a measure of its error-detection capability. An error-control code can be used to detect all patterns of u errors in any codeword as long as $d_{min}=u+1$. The code also may detect many error patterns with more than u errors, but it is guaranteed to detect all patterns of u errors or less. We'll assume that the error-detection decoder comprises a look-up table with all 2^k valid codewords stored. When an n-bit word is received by the decoder, it checks the look-up table, and if this word is one of the allowable codewords, it flags the n-bit word as error-free and sends the corresponding information bits to the user.

We'll use Figure 1 to illustrate three cases: no errors, a detectable error pattern, and an undetectable error pattern.

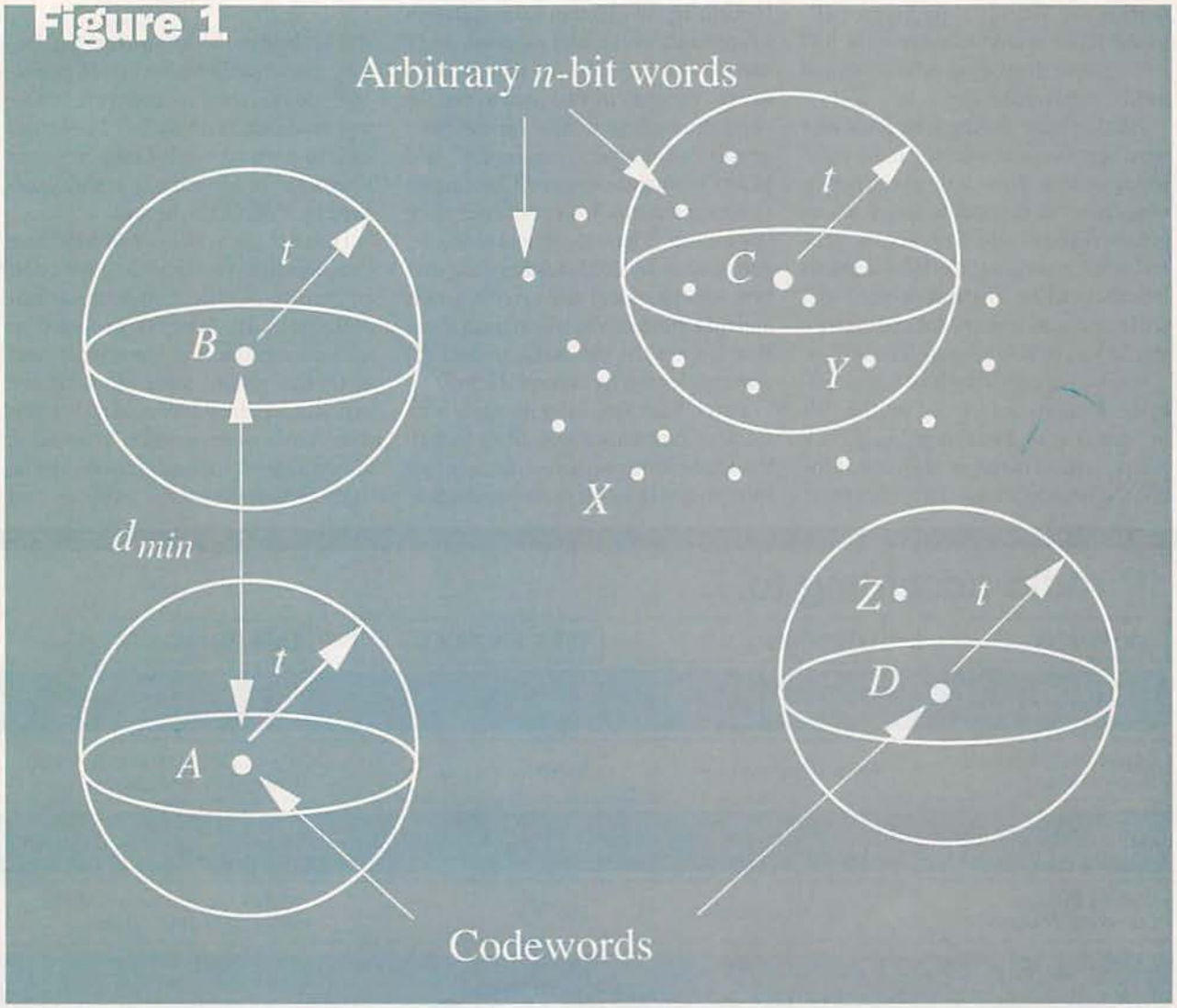
■ **No errors:** Assume the encoder sends codeword C, and the channel introduces no errors. Then codeword C also will be received, the decoder will find it in the look-up table and decoding will be successful.

■ **Detectable error pattern:** This time we send codeword C, and the channel introduces errors such that the n-bit word Y is received. Because Y is not a valid codeword, the decoder will not find it in the table and will therefore flag the received n-bit word as an errored codeword. The decoder does not necessarily know the number or location of the errors, but that's OK because we only asked the decoder to detect errors. Because the decoder properly detected an errored codeword, decoding is successful.

■ **Undetectable error pattern:** We send codeword C for the third time, and

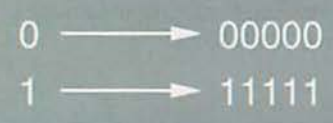
Continued on page 51

Figure 1



Continued from page 35

Figure 2



this time the channel introduces the unlikely (but certainly possible) error pattern that converts codeword *C* into codeword *D*. The decoder can't know that codeword *C* was sent and must assume that codeword *D* was sent instead. Because codeword *D* is a valid codeword, the decoder declares the received *n*-bit word error-free and passes the corresponding information bits on to the user. This is an example of decoder failure.

Naturally, we want the decoder to fail rarely, so we choose codes that have a small probability of undetected error. One of the most popular error-detection codes is the shortened Hamming code, also known as the cyclic redundancy check (CRC).

Comparing the spheres surrounding codewords *A* and *B* in Figure 1, we see that the error-correcting capability of a code is given by $d_{min}=2t+1$ (this is the minimum separation that prevents overlapping spheres). In other words, a code with $d_{min}=3$ can correct all patterns of 1 error, one with $d_{min}=5$ can correct all patterns of 2 errors, and so on. A code can be used to correct *t* errors and detect *v* additional errors as long as $d_{min} \geq 2t+v+1$. Now refer to Figure 1 and consider the following error-decoding cases: correct decoding, decoding failure and error detection without correction.

■ **Correct decoding:** Assume that codeword *C* is sent, and the *n*-bit word *Y* is received. Because *Y* is inside *C's* sphere, the decoder will correct all errors.

■ **Decoding failure:** This time we send codeword *C*, and the channel gives us *n*-bit word *Z*. The decoder has no way of knowing that codeword *C* was sent and must decode to *D* since *Z* is in *D's* sphere. This is an example of error-correction decoder failure.

■ **Error detection without correction:** This case shows one way that an error-correction code can be used to also detect errors. We send codeword *C* and receive *n*-bit word *X*. Since *X* is not inside any sphere, we won't try to correct it. We do, however, recognize that it is an errored codeword and report this information to the user.

Table 1

Post-Decoding Probability of Bit Error

(5,1) Repetition Code

Input BER	Output BER
10^{-2}	9.9×10^{-6}
10^{-3}	1.0×10^{-8}
10^{-4}	1.0×10^{-11}
10^{-5}	1.0×10^{-14}

In the last example, we could try to correct *n*-bit word *X* to the nearest valid codeword, even though *X* was not inside any codeword's sphere. A decoder that attempts to correct all received *n*-bit words regardless of whether they are in a decoding sphere is called a complete decoder. On the other hand, a decoder that attempts to correct only *n*-bit words that lie inside a decoding sphere is called an incomplete, or bounded-distance decoder. Bounded-distance decoders are much more common than complete decoders.

Now let's apply what we've learned to a simple error correction code, known as the repetition code. Consider a (5, 1) repetition code that repeats each bit four times. Figure 2 depicts such an encoder.

The decoder takes 5 bits at a time and counts the number of 1s. If there are three or more, the decoder selects 1 for the decoded bit. Otherwise, the decoder selects 0. The minimum distance of this code is 5, so it can correct all patterns of two errors. To compute the error performance of this code, consider a random error channel with a probability of bit error of *p*. After decoding, the probability of bit error is simply the probability of three or more bit errors in a 5-bit codeword. This probability is computed for several values of *p* with results listed in Table 1.

Next month: Error-control coding techniques. ■

Jay Jacobsmeyer is president of Pericle Communications Co., a consulting engineering firm located in Colorado Springs, Colo. He holds bachelor's and master's degrees in Electrical Engineering from Virginia Tech and Cornell University, respectively, and has more than 20 years experience as a radio frequency engineer.

References:

[1] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, pp. 379-423, 1948.

[2] R. E. Blahut, Theory and practice of error control codes, Reading, Massachusetts: Addison-Wesley Publishing Co., 1983.

[3] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," IEEE Transactions on Communications, pp. 1261-71, October 1996.

techspeak

BY JAY M. JACOBSMEYER, P.E.

Error-control coding options abound

Advanced techniques for modern land mobile radios offer plusses, minuses

Part 3 of 4

Last month we covered applications and limitations of error-control codes and some of the theory behind their operation. This month we cover coding techniques. Specifically, we will discuss automatic repeat request, or ARQ, forward error correction, hybrid ARQ, interleaving and concatenation.

An error-detection code by itself does not control errors, but it can be used to request repeated transmission of errored code words until they are received error-free. This technique is called ARQ. In terms of error performance, ARQ outperforms forward error correction (FEC) because code words always are delivered error-free (provided the error-detection code doesn't fail). However, this performance does not come free of charge—we pay for it with decreased throughput.

The chief advantage of ARQ is that error detection requires simpler decoding than error correction. ARQ also is adaptive because it only re-transmits information when errors occur. On the other hand, ARQ schemes require a feedback path that may not be available.

There are two types of ARQ:

■ **Stop-and-wait ARQ:** With stop-and-wait ARQ, the transmitter sends a single code word and waits for a positive acknowledgement (ACK) or negative acknowledgement (NAK) before sending any more code words. The advantage of stop-and-wait ARQ is that it only requires a half-duplex channel. The main disadvantage is that it wastes time waiting for ACKs, resulting in low throughput.

■ **Continuous ARQ:** Continuous ARQ requires a full duplex channel because code words are sent continuously until a NAK is received. Typically, only the code word corresponding to the NAK is re-transmitted. Continuous ARQ offers greater throughput efficiency than stop-and-wait ARQ at the cost of greater memory requirements.

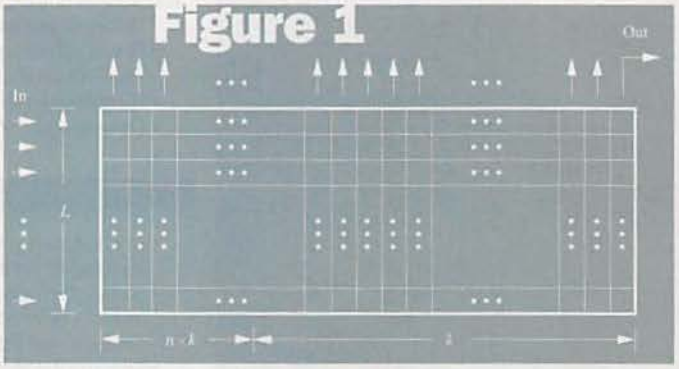
FEC is appropriate for applications where the user must get the message right the first time. A voice circuit is one example. Today's error-correction codes fall into two categories:

■ **Block codes:** The repetition code introduced in last month's column (*MRT*, January, page 51) is an example of a binary block code. It is important to note that not all block codes are binary. In fact, one of the most popular block codes is the Reed-Solomon code that operates on *m*-bit *symbols*, not bits. Because Reed-Solomon codes correct symbol errors rather than bit errors, they are very effective at correcting burst errors. For example, a 2-symbol error-correcting Reed-Solomon code with 8-bit symbols can correct all bursts of 16 bits or less in length. Reed-Solomon codes are used in JTIDS, a NASA deep-space standard, and CD players.

■ **Convolutional codes:** With convolutional codes, the incoming bit stream is applied to a *K*-bit-long shift register. For each modification to the shift register, *b* new bits are inserted and *n* code bits are delivered, so the code rate is *b*/*n*. The power of a convolutional code is a function of its constraint length, *K*. Large constraint-length codes tend to be more powerful.

Unfortunately, with large constraint length comes greater decoder complexity. There are several effective decoding algorithms for convolutional codes, but the most popular is the Viterbi algorithm, discovered by Andrew Viterbi in 1967.

One drawback of the codes we have examined so far is that they require bandwidth expansion to accommodate the added parity bits should the user wish to maintain the original unencoded information rate. In 1976, Gottfried Ungerboeck discovered a class of codes that integrates the encoding and modulation functions and does not require bandwidth expansion [1]. These codes are called Ungerboeck codes or trellis coded modulation (TCM). Nearly every telephone line modem or DSL modem on the market today operating above 9.6 kb/s uses TCM.



Other coding techniques exist that are worth examining, such as hybrid ARQ. Hybrid ARQ schemes combine error detection and FEC to make more efficient use of the channel. At the receiver, the decoder first attempts to correct any errors present in the received code word. If it cannot correct all the errors, it requests retransmission using an ARQ technique.

Another technique to consider is interleaving. Many real-world channels are burst-error channels. For example, the mobile radio channel is a burst-error channel as a consequence of multipath fading. The most popular way to correct burst errors is to take a code that works well on random errors (e.g., a convolu-

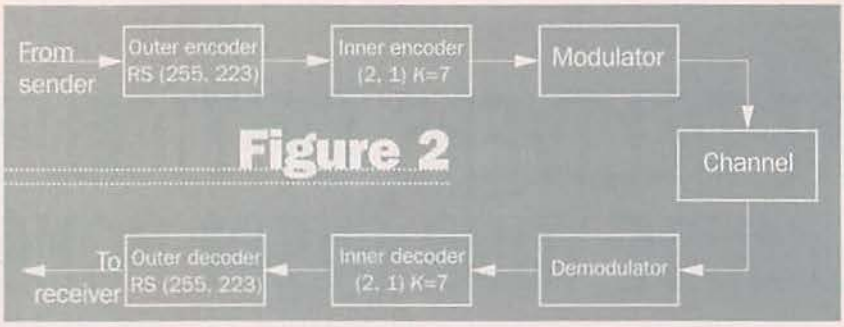


Figure 2

tional code) and interleave the bursts to "spread out" the errors so that they appear random to the decoder. There are two types of interleavers commonly in use today, block interleavers and convolutional interleavers. Figure 1 on page 38 illustrates the operation of a block interleaver.

The block interleaver is loaded row by row with L code words, each with a length of n bits. These L code words are then transmitted column by column until the interleaver is emptied. Then the interleaver is loaded again, and the cycle repeats. At the receiver, the code words are deinterleaved before they are decoded. A burst of length L bits or less will cause no more than 1 bit error in any single code word. The random error decoder is much more likely to correct this single error than the entire burst.

The parameter L is called the interleaver degree, or interleaver depth. The interleaver depth is chosen based on worst-case channel conditions. It must be large enough so that the interleaved code can handle the longest error bursts expected on the channel. The main drawback of interleavers is the delay introduced with each row-by-row fill of the interleaver. The delay is a function of the interleaver depth, which, in turn, is a function of the fade duration on the channel. The delay on some channels can be several seconds long. This long delay is often unacceptable. On voice circuits, for example, interleaver delays confuse the unfamiliar listener by introducing long pauses between speaker transitions. Even short delays of less than one-half second are sufficient to disrupt normal conversation.

In theory, interleaving is a poor way to handle burst errors. Why? From a strict probabilistic sense, we are converting "good" errors into "bad" errors. Burst errors have structure, and that structure can be exploited (in theory).

Strategy:
A plan of action to achieve a goal
• Compact size
• Best price
• Best performance in class
rfneulink
NL6000
It's your move
NL6000 VHF/UHF Radio Modem
www.rfneulink.com (800) 233-1728

TRANSMITTER LOCATION
New fixed site direction finders provide 2 degree accuracy, and include software for triangulation from a central control site. Mobile versions also available covering 50MHz to 1 GHz
Doppler Systems Inc.
PO Box 2780 Carefree, AZ 85377
Tel: (480) 488-9755 Fax: (480) 488-1295
www.dopsys.com
European Marketing Director Denis Egan
PO Box 2, Seaton, Devon EX12 2YS England
Tel & Fax: 44 1297 62 56 90

Interleavers "randomize" the errors and destroy the structure. Despite this theoretical disadvantage, interleaving is one of the best burst-error-correcting techniques in practice. In fact, the greatest advance in coding theory in the past 15 years, turbo coding, employs a very long random interleaver [2]. Until the coding theorists discover a better way, interleaving will be an essential error-control-coding technique for bursty channels.

Yet another technique worth exploring is concatenation. When two codes are used in series, the combination is called a concatenated code. Concatenated codes often are used when a single code cannot correct all types of errors encountered on the channel. The operation of concatenated codes is best illustrated by example. Figure 2 on page 40 shows the Consultative Committee for Space Data Systems (CCSDS) Blue Book standard for Telemetry Channel Coding (interleaving is omitted for clarity) [3].

The inner code—a rate $^1/_2$, K=7, convolutional code with Viterbi decoding—corrects most random errors and the outer code—a 16-symbol error correcting Reed-Solomon code—cleans any burst errors that slip through the Viterbi decoder. The Reed-Solomon code operates on 8-bit symbols and therefore is a very powerful burst-error correcting code. The overall code rate is simply the product of the two individual code rates, i.e., $(^1/_2)(223/255) \approx 0.44$. It should be noted that coding gains for concatenated codes are very large.

Next month: Error-control coding in APCO 25 Radios. ■

Jay Jacobsmeyer is president of Pericle Communications Co., a consulting engineering firm located in Colorado Springs, Colo. He holds bachelor's and master's degrees in Electrical Engineering from Virginia Tech and Cornell University, respectively, and has more than 20 years experience as a radio frequency engineer.

References:

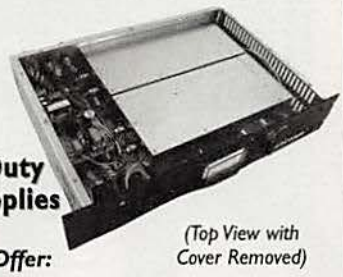
[1] G. Ungerboeck, "Trellis coded modulation with redundant signal sets Parts I and II," IEEE Communications Magazine, vol. 25, No. 2, pp. 5-21, February 1987.

[2] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," IEEE Transactions on Communications, pp. 1261-71, October 1996.

[3] E. R. Berlekamp et al., "The application of error control to communications," IEEE Communications Magazine, vol. 25, No. 4, pp. 44-57, April 1987.

DuraComm®
POWER SUPPLIES

NEW



RLP Series
Commercial Heavy Duty
AC to DC Power Supplies

(Top View with Cover Removed)

These Rack Mount Units Offer:

- Power Factor Correction (PFC)
- Universal Input Voltage 90-264VAC Automatic Sensing
- Overload, Over Voltage & Over Temperature Protections
- Remote Power Control
- 13" Deep, Ventilated, Gold Irridite Aluminum Chassis

With Optional:

- Battery Back-Up
- Battery Charging
- Smart Charging
- Low Voltage Disconnect

Visit us on the web at www.duracomm.com
or email us sales@duracomm.com

Phone 1•800•467•6741 Fax 1•800•825•1403

203 W. 23rd Ave. North Kansas City, MO. 64116

YOU Told Us What You Wanted

COM3010
Communications Service Monitor

- ✔ 100kHz - 1.0GHz, Full Duplex!
- ✔ ±0.1ppm Frequency Accuracy!
- ✔ Built-in Power Meter & Load!
- ✔ Built-In Frequency Counter!
- ✔ Built-In Sweep Generator!
- ✔ Built-In SINAD Meter!
- ✔ Built-In Calibrated RSSI Meter!
- ✔ Built-In RS232 Control
- ✔ Built-In Li-Ion Battery Pack

As Low As $3995!

STE Series
RF Shielded Test Enclosures

- ✔ Wireless Device Testing
- ✔ Cellular, PCS, GSM Testing
- ✔ Bluetooth, RFID, 3G Testing
- ✔ 802.11a, b, g Testing
- ✔ RF Radiation Test Fixture Models
- ✔ Super Isolation Up To 18 GHz!
- ✔ Stock Models To Fit Your Custom Applications!
- ✔ Variety of I/O Options

As Low As $439!

We Listened...Consider It Done!

ramsey electronics inc.

RAMSEY ELECTRONICS, INC.
590 Fishers Station Drive
Victor, NY 14564
(585) 924-4560
www.ramseytest.com

For decades, your leader in cost effective, high performance RF Test Equipment!

Specifications, prices, and lease terms subject to change. Contact Ramsey for additional information.

techspeak

BY JAY M. JACOBSMEYER, P.E.

Error-control coding for P25 radios

Adapt coding schemes to channel characteristics to maximize performance

Part 4 of 4

Project 25 is a digital radio airlink standard designed to create interoperability on 12.5 kHz and 6.25 kHz channels. Although Project 25 radios can pass voice or data communications, this article will focus strictly on error-control coding for voice communications.

P25 modulation comes in two flavors, Compatible 4 FM (C4FM) and Compatible Quadrature Phase Shift Keying (CQPSK) [1]. C4FM is intended for Phase I P25 systems that operate on 12.5 kHz channels, while CQPSK is intended for Phase II P25 systems that will operate on 6.25 kHz channels. In both cases, the signaling rate is 4.8 k symbols per second (sps) and there are two bits per symbol for a gross bit rate of 9.6 kb/s.

The vocoder (voice encoder/decoder) adopted by the APCO Project 25 committee is the Improved Multi-Band Excitation (IMBE) vocoder, developed by Digital Voice Systems Inc. [2]. The IMBE vocoder operates at a basic rate of 4.4 kb/s, but an additional 2.8 kb/s is required for error-control coding. Thus, the bit rate for the vocoder and its error-control coding is 7.2 kb/s. An additional 2.4 kb/s is used for channel signaling, so the gross bit rate is 9.6 kb/s.

Rather than think in terms of bit rate, we also can think in terms of bits per vocoder frame. Each frame is 20 milliseconds (ms) long (50 frames per second) and contains 88 vocoder bits, 56 error-control-coding bits and 48 signaling bits for a total of 192 bits.

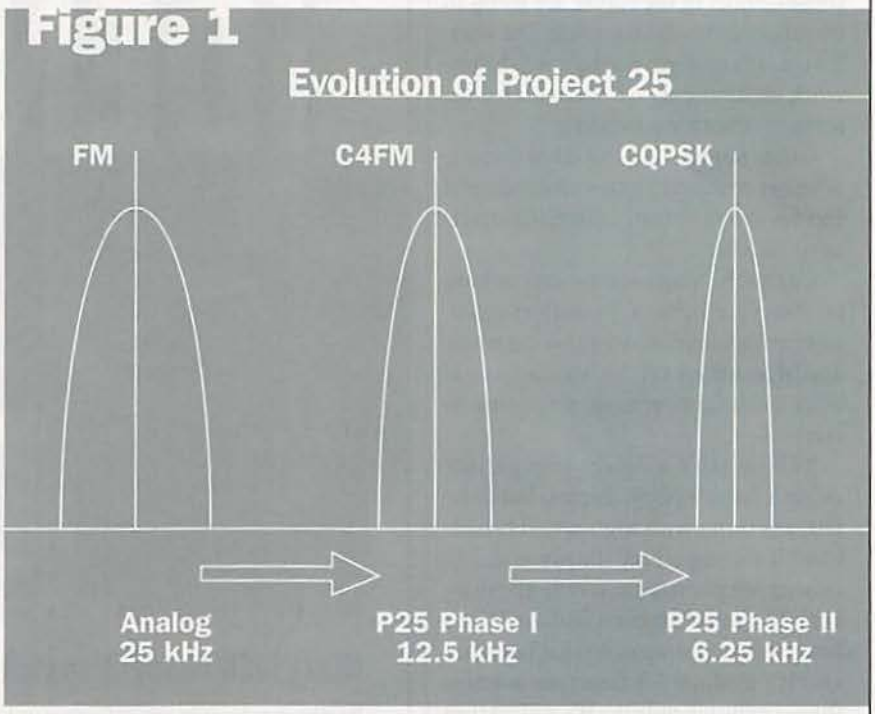
A full explanation of the IMBE vocoder is beyond the scope of this article, but suffice it to say that unlike most cell phone vocoders, IMBE was optimized for low bit-rate applications (less than 4.8 kb/s). However, like most cell phone vocoders, IMBE is model-based, meaning that rather than sending digital samples of voice, the vocoder sends the parameters to be used in a human speech model.

Fewer bits are needed to specify the model parameters, and both the transmitter and the receiver include a full description of the model before they leave the factory. The IMBE vocoder and its successor, AMBE, also are used in many satellite voice networks, including Iridium's.

The mobile radio channel is hostile to digital communications, and for best performance, the error-control-coding scheme must be adapted to the channel's characteristics. Because of multipath fading, the mobile radio channel is inherently a burst-error channel, and compatible correcting techniques should be employed. Candidate burst-error-correcting techniques include Reed-Solomon codes and interleaving (which can be used with any code).

The protection provided by the 56 error-control-coding bits is not applied equally over the 88 vocoder bits. Some vocoder bits are more important for intelligibility than others, so nearly all modern vocoders, including IMBE, apply error-control coding unequally. IMBE uses three principal coding techniques: interleaving, Hamming codes, and Golay codes. The signaling protocol (56 bits per frame) employs other coding techniques, including Reed-Solomon codes and trellis coded modulation. We'll limit the following discussion to the IMBE techniques: interleaving, Hamming and Golay codes.

Figure 1



Bit interleaving is used to spread short bursts of errors among several code words. The error-correction decoder is less likely to be overwhelmed if these bursts of errors arrive in multiple code blocks. The

Figure 2 Bit arrangement in 20 ms vocoder frame

After encoding: 144 bits/frame x 50 frame/s=7.2 kb/s

frame is interleaved in such a way that the minimum separation between any two bits of the same error-correction code is 3 symbols, or 6 bits [2]. Long bursts, such as those that occur when the user is walking or driving slowly, will be longer than the interleaver depth, so the interleaver design is not perfect. The designer is forced to make tradeoffs because greater interleaver depth means longer delay, and very long delays cannot be tolerated, especially on voice networks.

The 56 error-control bits are divided between four (23, 12) Golay codes and three (15, 11) Hamming codes. Both codes are binary block codes. Recall that an (n, k) binary block code has k information bits and n-k parity bits. In our case, there are 4x11 parity bits for the Golay codes plus 3x4 parity bits for the Hamming codes, for a total of 56 error-control bits. Figure 2 shows the arrangement of bits in the vocoder frame.

Hamming codes can correct just one error per code word. They often are used in a shortened form in other communication systems for error detection and are called cyclic redundancy checks, but in P25, the Hamming code is used strictly for error correction. The Hamming code is a single error-correcting perfect code.

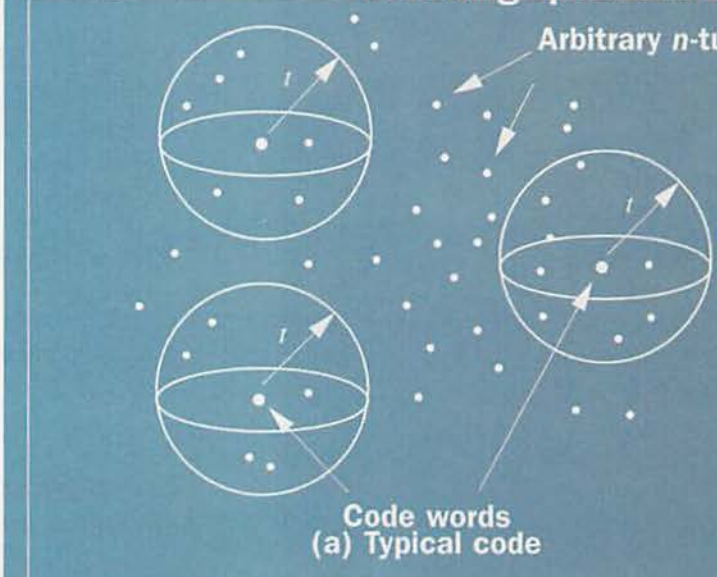
The (23, 12) Golay code has some unique properties. It is the only known multiple-error-correcting binary perfect code. It can correct any combination of three or fewer random errors in a block of 23 bits.

So, what is a perfect code? Recall from January's Tech Speak that we can represent a code word in decoding space as the center of a sphere of radius t, where t is the number of errors the code can correct. The encoder takes k bits at time and encodes them into an n-bit code word. For a binary code, there are 2^k code words, but there are 2^n possible n-tuples to consider at the decoder because bit errors may occur on the channel.

For nearly all codes, many of the n-tuples will fall outside any sphere, and this information can be used to detect an errored code word even if it cannot be corrected. For the very small class of perfect codes (there are only two nontrivial binary perfect codes), all n-tuples fall within a decoding sphere. This relationship is shown in Figure 3.

If this were a data system, the use of error-control codes would be relatively straightforward. Some number of parity bits would be dedicated to error detection to ensure a small probability of undetected errors. The remaining parity bits would be used for error correction. Those frames with uncorrectable error patterns (as determined by the error-detection code) would be re-

Figure 3 Decoding spheres for most



Equation 1

$$\varepsilon_T = \sum_{i=0}^{6} \varepsilon_i$$

Equation 2

$$\varepsilon_R = (0) = .95\varepsilon_R(-1) + .000365\varepsilon_T$$

where $\varepsilon_R(0)$ is the estimate of the error rate for the current frame and $\varepsilon_R(-1)$ is the estimate for the previous frame.

transmitted using some form of automatic repeat request (ARQ), an error-correcting technique that always delivers error-free code words, but at the cost of decreased throughput.

Voice systems are different. Because voice quality is subjective, we have another bag of tricks to mitigate the effect of bit errors.

The IMBE vocoder performs error estimation as part of the decoding process. The speech decoder estimates the number of errors in each received frame by computing the number of errors corrected by each of the (23, 12) Golay codes and each of the (15, 11) Hamming codes (a total of 7 code words per frame). The expression for the error estimate is found in reference [2] on page 45 or in Equations 1 and 2. It includes an estimate of the number of errors in the current frame plus a weighted version of the error estimate of the previous frame. Thus, the algorithm looks back at earlier frames and has some ability to detect trends.

When we say the number of errors corrected, we really mean the number of errors the decoder thinks it corrected. For example, if the received Golay n-tuple differs from the nearest valid code word in 1 bit position, the decoder must assume there is just one error. However, the code word may actually have five errors, placing it in an entirely different decoding sphere. This is an example of decoding failure, and it affects the accuracy of the error estimate. When the error rate is high, the probability of decoding failure also is high, and the error estimate will be somewhat flawed.

Once the error estimate is computed, the decoder does the following:

1. If the error estimate is below some threshold, it decodes the frame normally.

2. If the error estimate is above the threshold, it:

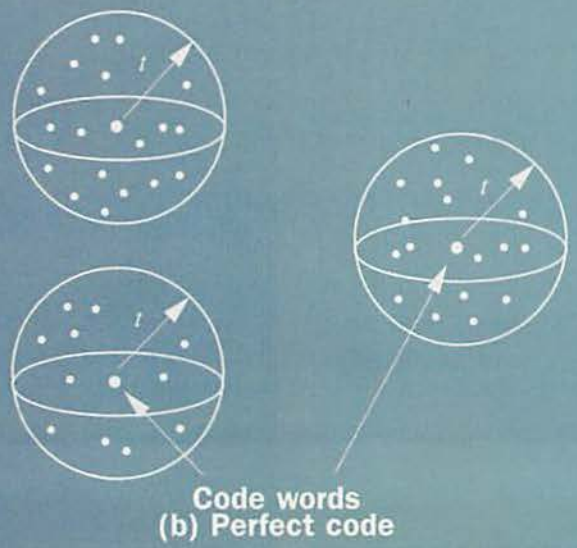
- repeats the frame,
- mutes the frame or
- performs adaptive smoothing of the frame.

When a frame is repeated, the assumption is that it is better to "stretch" the speech fragment rather than deliver a frame that is known to have problems. Remember that a frame is only 20 ms long, so a single repeat may not be noticeable to the human ear.

In a severe bit-error environment, a frame is muted. When muted, a replica of white noise is created, sometimes called "comfort noise."

Adaptive smoothing allows the decoder to take information from the previous frame and the next frame (there is a two-frame look-ahead in IMBE) to "smooth" the effects of the errored frame. Adaptive smoothing only is done over an intermediate range of bit error estimates. ■

codes and perfect codes



Jay Jacobsmeyer is president of Pericle Communications Co., a consulting engineering firm located in Colorado Springs, Colo. He holds bachelor's and master's degrees in Electrical Engineering from Virginia Tech and Cornell University, respectively, and has more than 20 years experience as a radio frequency engineer.

References:

[1] TIA-102.BAAA-A, "Project 25 FDMA: Common Air Interface," Sept. 2003.

[2] TIA-102.BABA, "Project 25 Vocoder Description," Dec. 2003.

[3] S. Lin and D.J. Costello Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, 1983.